Práctica 2: Odoo como ERP

Eder Martínez Castro

17 de noviembre de 2025

Índice general

1. Objetivo 2
2. Activación del modo desarrollador 2
3. Implementación del Módulo (Hola mundo) 4
4. Implementación de Módulo (Lista de tareas) 6
   - 4.1. Crear la estructura de un módulo 6
   - 4.2. Lógica de nuestro módulo 8
   - 4.3. Comprobación del módulo creado 11
   - 4.4. Funcionamiento del módulo 12
5. Mejoras en el Módulo de (Listea de tareas) 14
6. Conclusión 16

1. Objetivo

El objetivo de esta práctica es conseguir implantar y configurar en nuestro Odoo los siguientes módulos **"Hola mundo"** y **"Lista de tareas"**, y para finalizar mejoraremos el módulo de **"Lista de tareas"** en el modelo y la vista.

Con esta práctica se busca tener el primer contacto con la creación de módulos en Odoo para poder sacarle mucho más provecho y personalización a nuestro Odoo.

2. Activación del modo desarrollador

Para poder activar el modo desarrollador en nuestro Odoo lo que tendremos que hacer es lo siguiente, tendremos que hacer el siguiente comando `"docker compose up -d"` si no tenemos nuestro contenedor corriendo, cuando ya lo tengamos levantado lo que haremos será irnos a nuestro navegador favorito y poner la ruta en la que tenemos desplegado nuestro proyecto en mi caso `http://localhost:8069`.

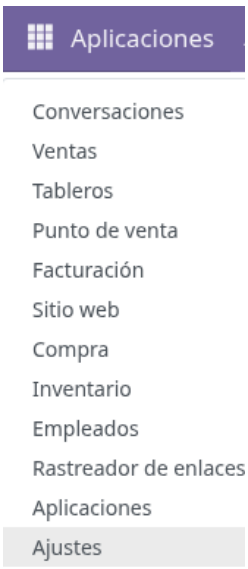

Figura 1: Como ir a ajustes de Odoo

Ya cuando estemos dentro de nuestro Odoo, nos dirigiremos a donde está el icono cuadrado en la cabecera morada, al hacer clic en este icono se nos abrirá una lista de opciones y seleccionaremos ajustes.

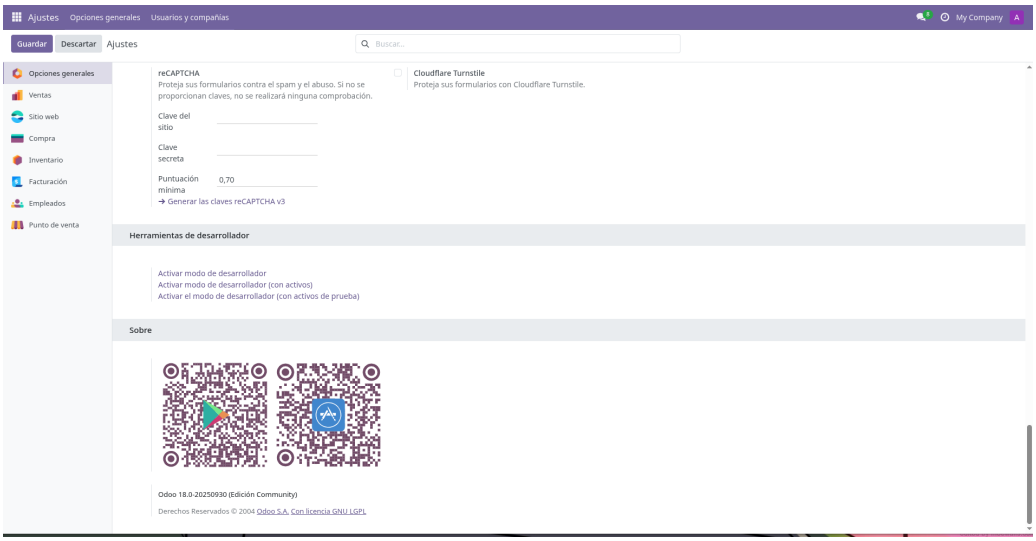

Figura 2: Modo desarrollador sin activar

Ahora que ya estamos en ajustes, nos situaremos en la sección de **opciones generales** y haremos scroll hasta encontrar la sección de **herramientas de desarrollador** en clicaremos en la opción de `activar modo de desarrollador` y se nos actualizará automáticamente.

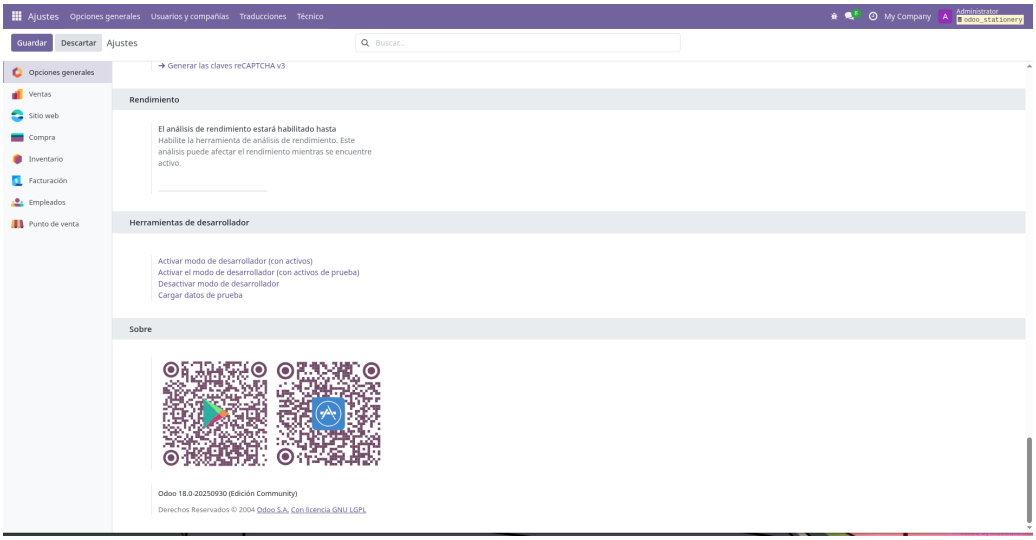

Figura 3: Modo desarrollador ya activado

Como podemos ver en esta captura podemos ver que ya tenemos el modo desarrollador activo y que tenemos la opción de desactivarlo si quisiéramos.

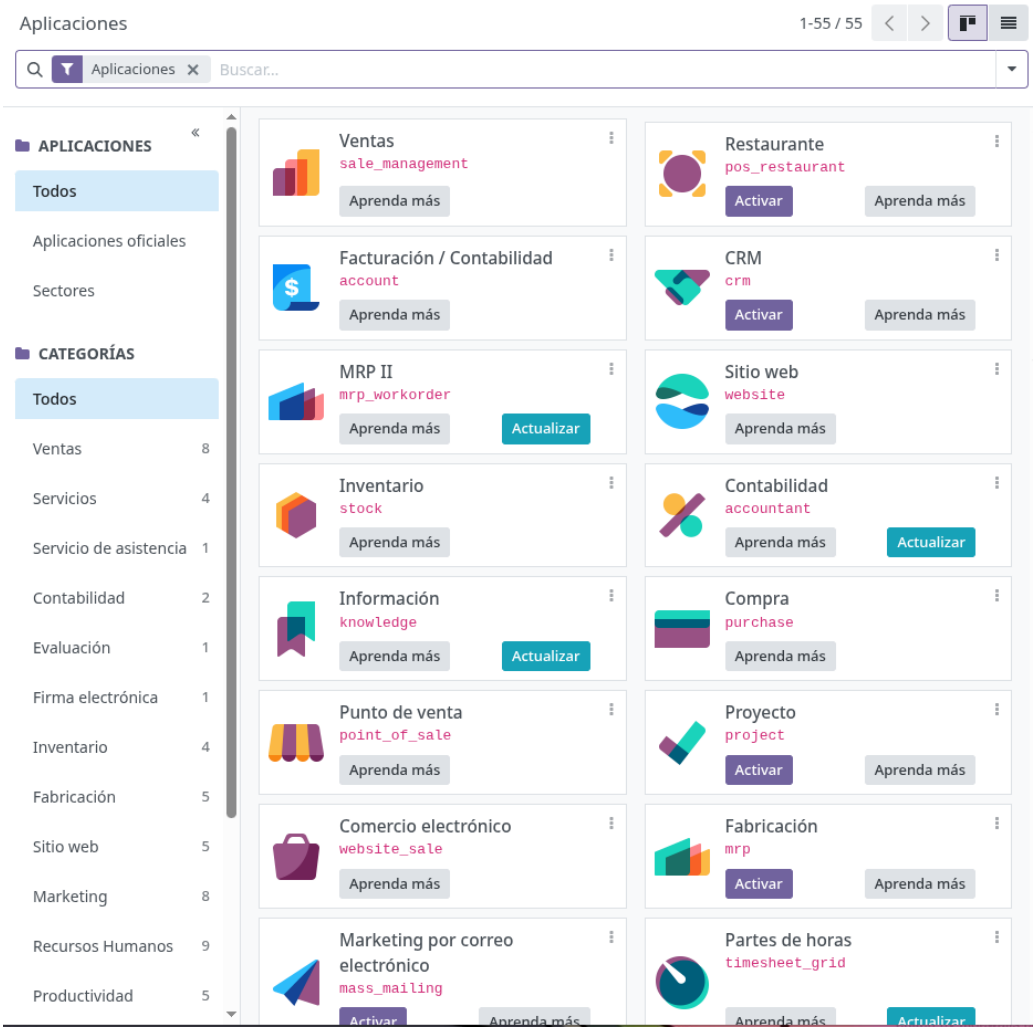

Figura 4: Cambios al estar en el modo desarrollador

Cuando tenemos el modo desarrollador activado el cambio más notable ya que se aprecia que ya no nos sale la descripción del módulo, si no que nos sale el como se llama la carpeta que aloja el módulo.

3. Implementación del Módulo (Hola mundo)

Ahora nos dirigimos a nuestro **IDE**, y abriremos nuestro proyecto donde tenemos nuestro **Odoo dockerizado**, ahora vamos a ver como crear nuestro primer módulo de una manera muy simple, ya que para este solo necesitaremos **crear estos 2 archivos**.

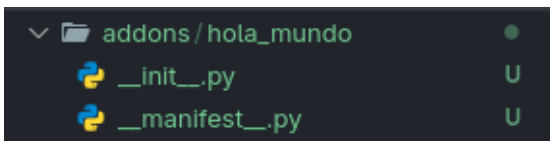

Figura 5: Así quedaría la estructura mínima para crear nuestro módulo hola_mundo

Explicación de los **archivos del módulo hola_mundo**:

- **___init___.py**: En este archivo se indica que nuestro módulo se puede cargar en Odoo.
- **___manifest___.py**: En este archivo tenemos la información básica de nuestro módulo para que Odoo la reconozca.

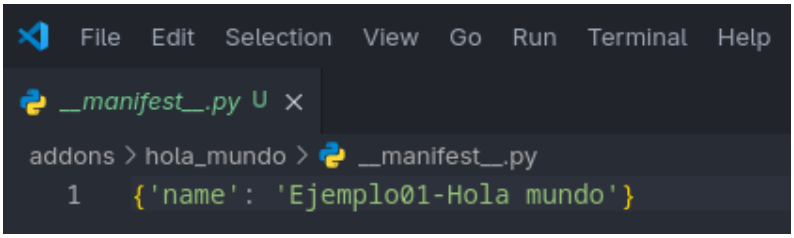

Figura 6: Contenido del archivo de ___manifest___.py

Solo necesitamos escribir una línea de código para poder mostrar nuestro módulo en Odoo obviamente no tendrá funcionalidad ninguna pero nos sirve para mostrar lo mínimo que se necesita para crear un módulo y que se muestre en las aplicaciones de Odoo.

Figura 7: Seleccionar actualizar lista de apliciones en el menú

Para actualizar la lista de las aplicaciones de nuestro Odoo, iremos al menú y seleccionaremos la opción de **actualizar lista de aplicaciones** que como podemos apreciar es la tercera opción de nuestro menú.

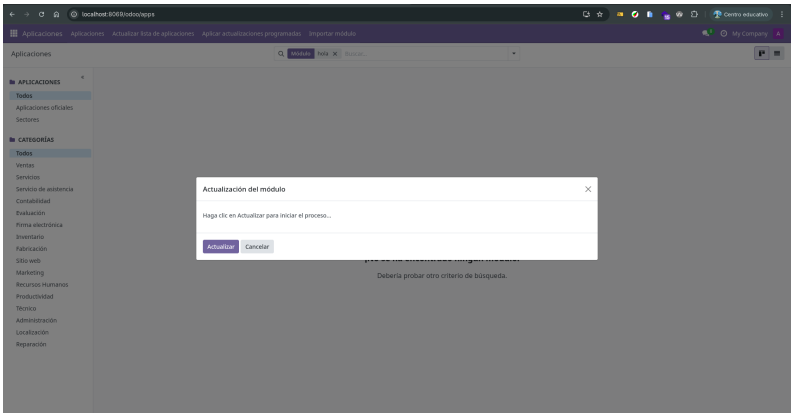

Figura 8: Actualizar la lista de apliciones para ver el nuevo módulo

Al clicar en **actualizar lista de aplicaciones** nos aparecerá esta ventana emergente, donde seleccionaremos el botón que pone **Actualizar**.

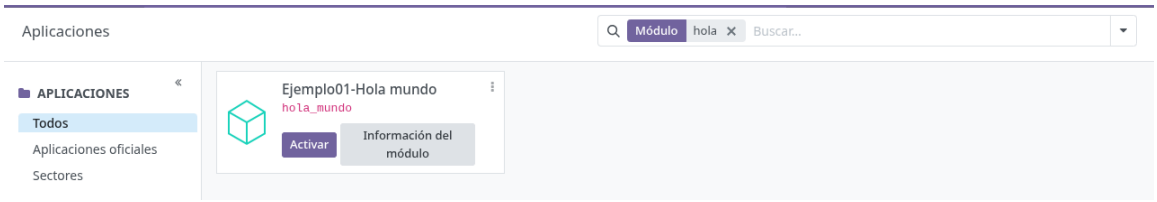

Figura 9: Módulo Ejemplo01-Hola mundo creado correctamente

Después de haber actualizado, nos dirigiremos al buscador donde buscaremos el módulo **hola_mundo**, y nos debería salir el módulo que acabamos de crear como se puede apreciar en esta imagen.

4. Implementación de Módulo (Lista de tareas)

Ahora vamos a empezar con la **creación de un módulo pero este con funcionalidad** donde veremos los pasos para poder crear un módulo sobre una lista de tareas sencilla, para gestionar las tareas que tengamos que hacer,

4.1. Crear la estructura de un módulo

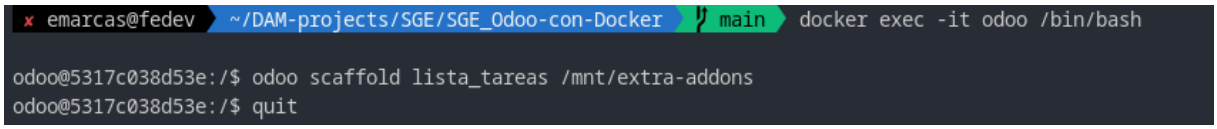

Figura 10: Comando para crear el módulo lista de tareas de Odoo

Para poder crear nuestro módulo emplearemos el comando **scaffold** y para ello como estamos trabajando con Odoo desde **Docker** lo primero que deberemos hacer es ejecutar los siguientes comandos.

Comando para entrar en nuestro contenedor de docker

```
docker exec -it <CONTAINER> /bin/bash
```

Comando para crear la estructura básica de un nuevo módulo en Odoo

```
odoo scaffold <MODULO_NAME> /mnt/extra-addons
```

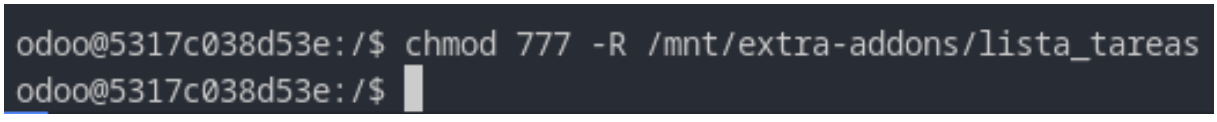

Figura 11: Darle permisos al módulo

Le vamos a **dar permisos a nuestro módulo** para poder trabajar más cómodamente con él, como por ejemplo permitirnos guardar los cambios que le vayamos haciendo a nuestro módulo.

Comando darle permisos a nuestro módulo

```
chmod 777 -R /mnt/extra-addons/<MODULO-NAME>
```

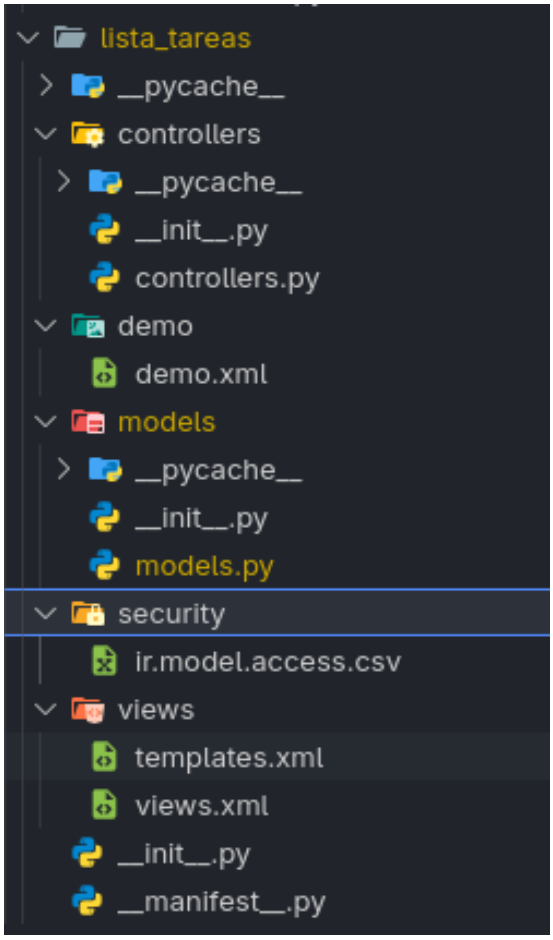

Figura 12: Comprobación de la estructura de nuestro módulo de Odoo creado mediante el comando **scaffold**.

4.2. Lógica de nuestro módulo

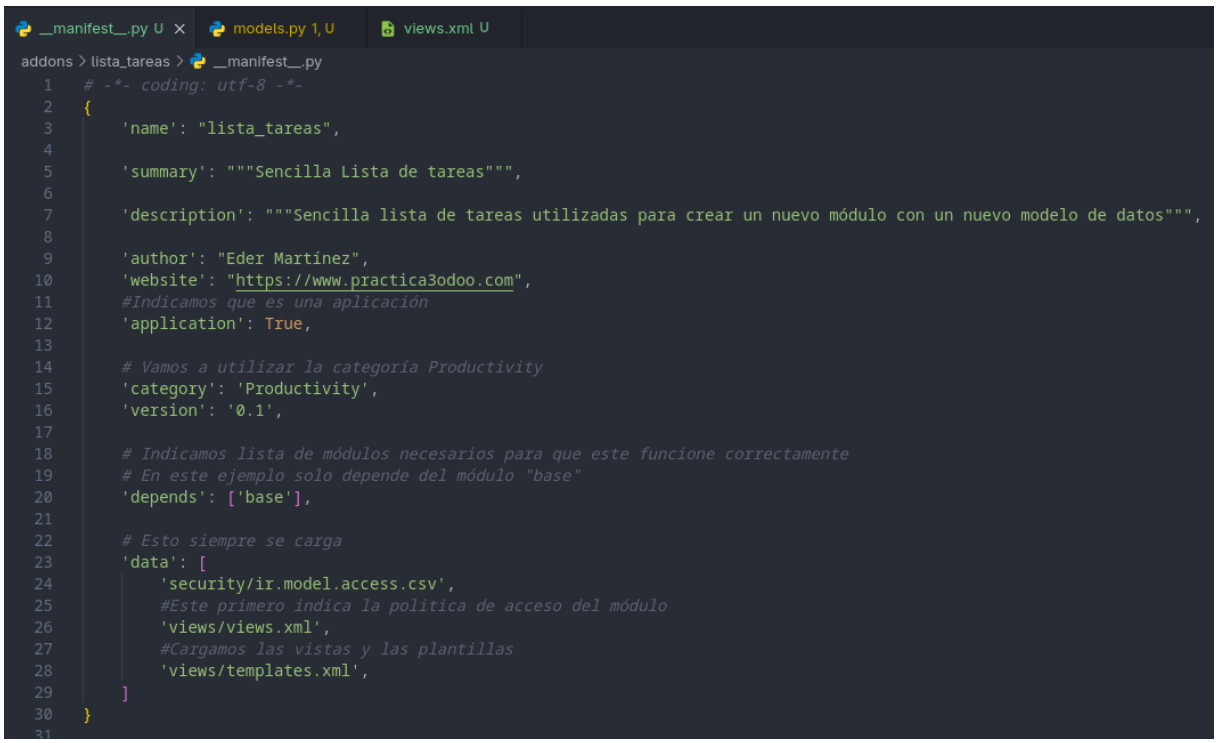

Figura 13: Archivo ___manifest___.xml

Ahora pegaremos el archivo `__manifest__.xml` este archivo es que **define y registra tu módulo** de nuestro módulo en Odoo, aquí es importante el `'application': True` que nos permitira que nuestro mádulo salga en el filtro de aplicaciones además de otras configuraciones extra.

__manifest__.py U models.py 1, U views.xml U

addons > lista_tareas > views > views.xml

```xml
<odoo>
  <data>
    <!-- explicit list view definition -->
    <!-- Definimos como es la vista explicita de la litas-->
    <record model="ir.ui.view" id="lista_tareas.list">
      <field name="name">lista_tareas list</field>
      <field name="model">lista_tareas.lista_tareas</field>
      <field name="arch" type="xml">
        <list>
          <field name="tarea"/>
          <field name="prioridad"/>
          <field name="urgente"/>
          <field name="realizada"/>
        </list>
      </field>
    </record>
    <!-- actions opening views on models -->
    <!-- Acciones al abrir las vistas en los modelos
    https://www.odoo.com/documentation/17.0/developer/reference/addons/actions.html
    -->
    <record model="ir.actions.act_window" id="lista_tareas.action_window">
      <field name="name">Listado de tareas pendientes</field>
      <field name="res_model">lista_tareas.lista_tareas</field>
      <field name="view_mode">list,form</field>
    </record>

    <!-- Top menu item -->
    <menuitem name="Listado de tareas" id="lista_tareas.menu_root"/>

    <!-- menu categories -->
    <menuitem name="Opciones Lista Tareas" id="lista_tareas.menu_1" parent="lista_tareas.menu_root"/>

    <!-- actions -->
    <menuitem name="Mostrar lista" id="lista_tareas.menu_1_list" parent="lista_tareas.menu_1" action="lista_tareas.action_window"/>
  </data>
</odoo>
```

Figura 14: Archivo views.xml

Ahora nos toca copiar el archivo `views.xml` que nos pasarón de ejemplo pero esto es para odoo:17 y tenemos que modificar un pequeño detalle ya no se utiliza **tree** para odoo:18 ahora se emplea **list**, con esta pequeña modificación ya lo arreglamos. Este archivo es el encargado de **configurar la interfaz** de nuestro módulo.

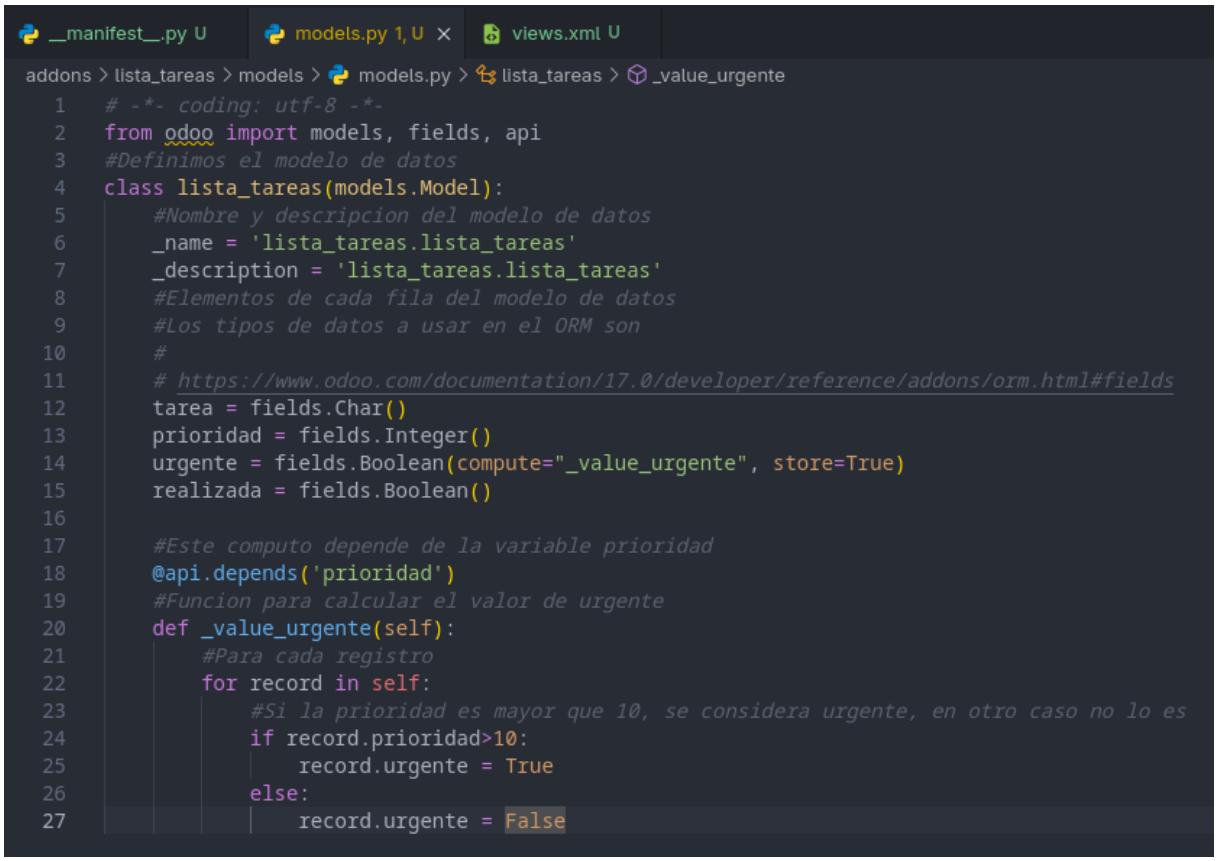

Figura 15: Archivo models.py

Por último pegaremos el archivo `models.xml` este archivo es el encargado de definir el **modelo de datos** de nuestro módulo de Odoo.

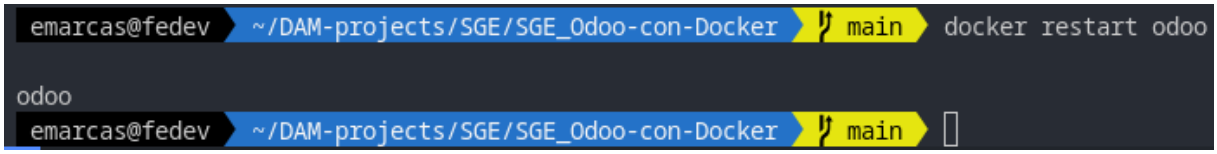

Figura 16: Reiniciar odoo para aplicar los cambios

Cuando ya hayamos modificado los dos archivos anteriores, lo que vamos a hacer es ejecutar el siguiente comando para reiniciar nuestro contenedor y asegurarnos de tenerlo actualizado y así evitarnos posibles problemas.

Comando para reiniciar nuestro contenedor y aplicar los cambios

```
docker restart <CONTAINER>
```

4.3. Comprobación del módulo creado

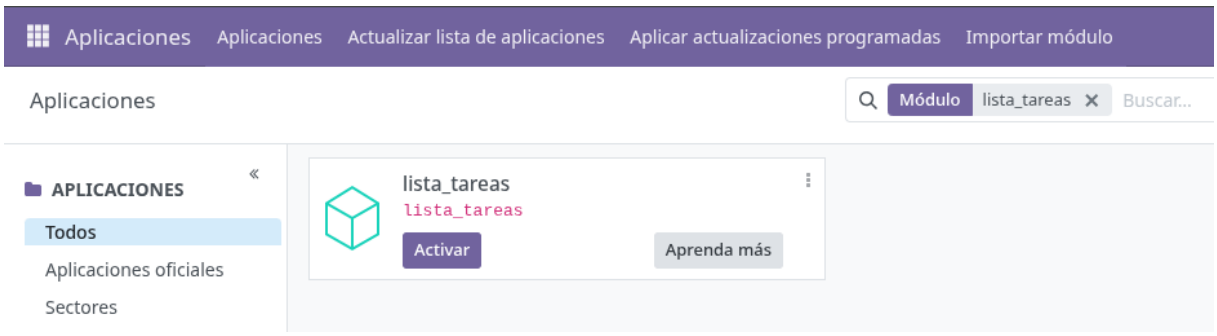

Figura 17: Módulo lista_tareas creado correctamente

Después de haber modificado los archivos anteriores y reiniciado nuestro contenedor lo que vamos a hacer es lo siguiente, nos dirigiremos a las aplicaciones de Odoo y buscaremos el módulo que acabmos de crear, así que buscaremos **lista_tareas** y procederemos a activar esta aplicación.

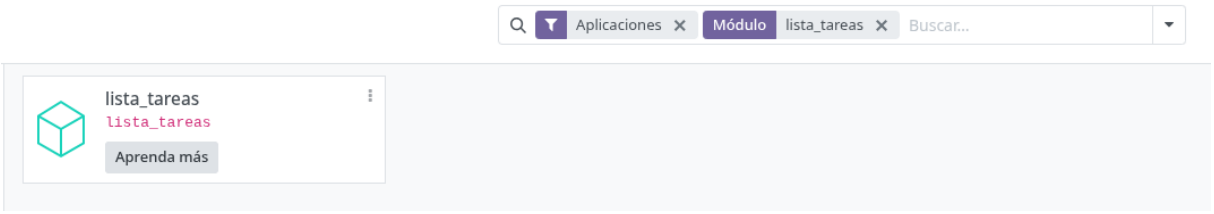

Figura 18: Módulo ya activado

Como podemos ver ya tenemos nuestro módulo **lista_tareas** ya activo ya listo para usarlo.

Figura 19: Entraremos en la aplicación de listado de tareas

Como ya tenemos activo nuestro módulo ahora nos dirigiremos al icono de aplicaciones y seleccionaremos nuestra aplicación **Listado de tareas** y pasaremos al siguiente apartado donde probaremos nuestro módulo.

4.4. Funcionamiento del módulo

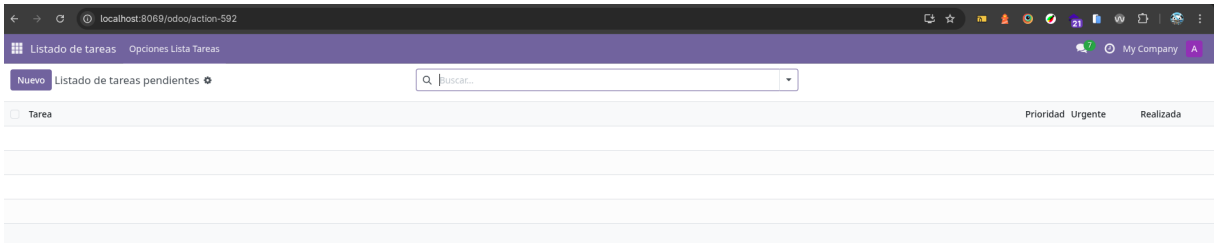

Figura 20: Nuestro módulo **listado de tareas** sin ninguna tarea porque todavía no añadimos ninguna todavía.

Al entrar en nuestra aplicación **Listado de tareas** podemos ver que es una interfaz bastante simple y sin color, que cuenta con los siguientes campos para nuestras tareas: tarea, prioridad, urgente y realizada.

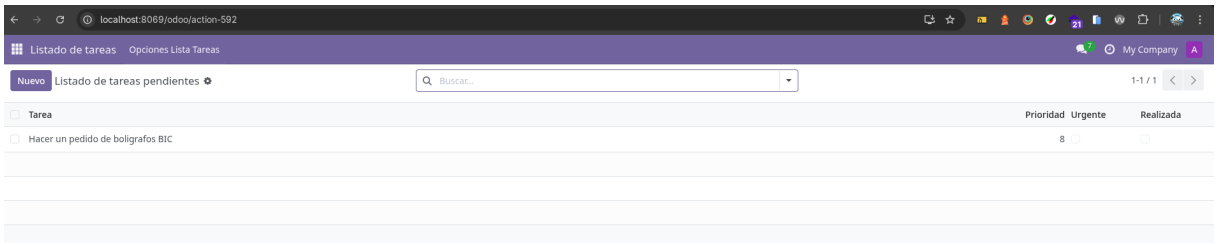

Figura 21: Agregamos una tarea a nuestro módulo **listado de tareas** en este caso una no urgente.

Al crear una nueva tarea **dependiendo del valor** ingresado en el campo prioridad **se marcará como urgente o no**, en este caso le pusimos de valor 8 y como no es >10 pues no se marca como urgente, luego también podemos clicar el checkbox para marcarla como completada o no.

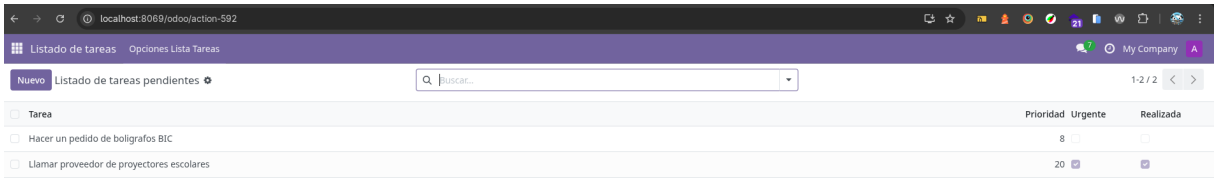

Figura 22: Agregamos otra tarea a nuestro módulo **listado de tareas** como prueba adicional.

En esta nueva tarea podemos ver como le di el valor de 20 y **ahora nos sale marcado el valor de urgente**, además para probar también la marqué como realizada para mostrar como se vería.

Como se puede apreciar **nuestro módulo cumple su función y funciona correctamente**, pero a continuación lo modificaremos para que por lo menos se vea más bonito y le agregaremos funcionalidad que realmente pueden ser muy útiles.

5. Mejoras en el Módulo de (Listea de tareas)

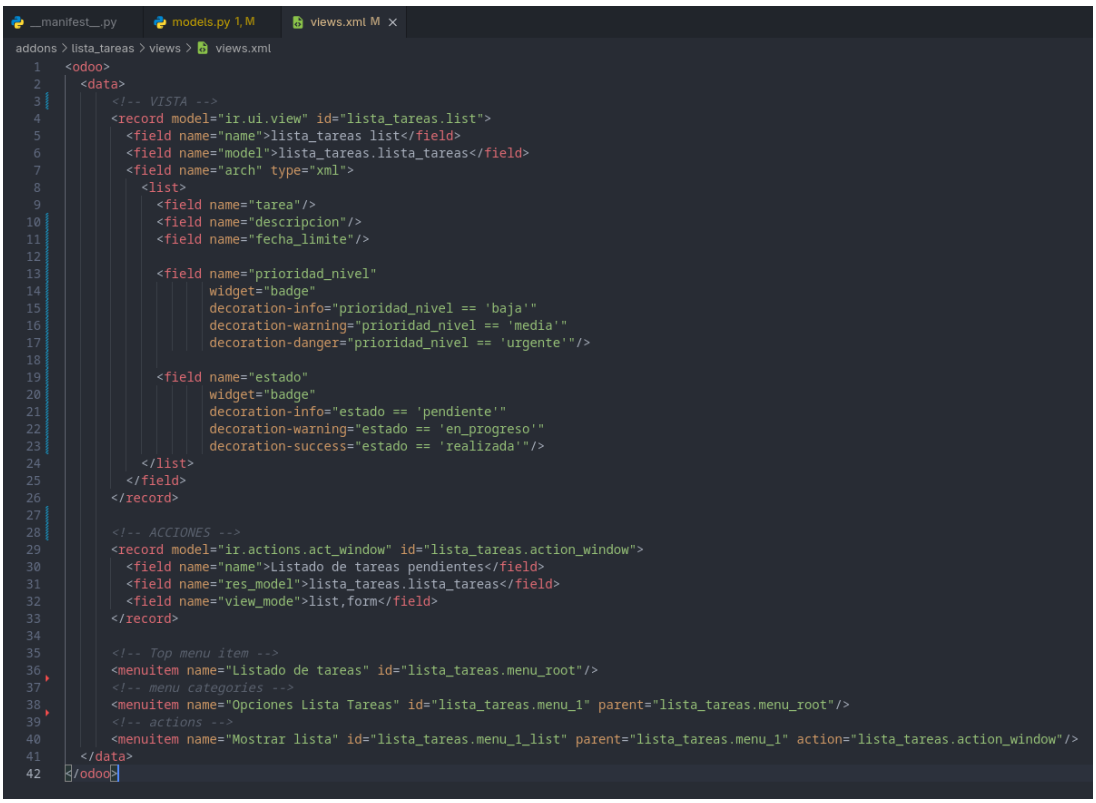

Figura 23: Modificación del archivo views.xml

Ahora vamos a explicar los cambios realizados en `models.py`, agregamos los **nuevos campos** para: descripción, fecha_limite, prioridad_nivel y estado. Y se agregaron **badges** para darle colores a prioridad_nivel y estado.

Figura 24: Modificación del archivo models.py

Ahora vamos a explicar los cambios realizados en `models.py`, lo primero fue agregar los **nuevos campos**: descripción, fecha_limite, prioridad_nivel y estado. **Se eliminó el campo urgente** y se reemplazó con el campo prioridad que ahora detecta tareas de prioridad: baja, media y urgente.

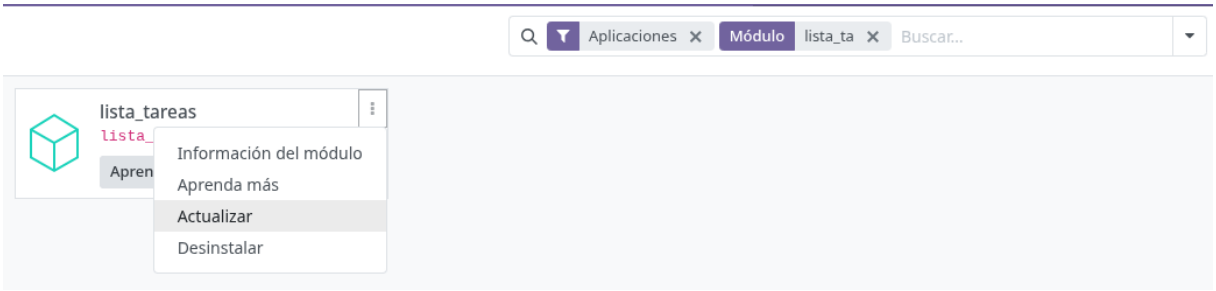

Figura 25: Actualizar nuestro módulo

Para poder obtener los cambios que hicimos para nuestro módulo, buscaremos nuestro **módulo lista_tareas** y haremos click en los tres puntos y seleccionaremos la opción de **actualizar**. Si esto no funcionará te recomiendo que vuelvas a reiniciar tu contenedor con el comando: `docker restart <CONTAINER>`.

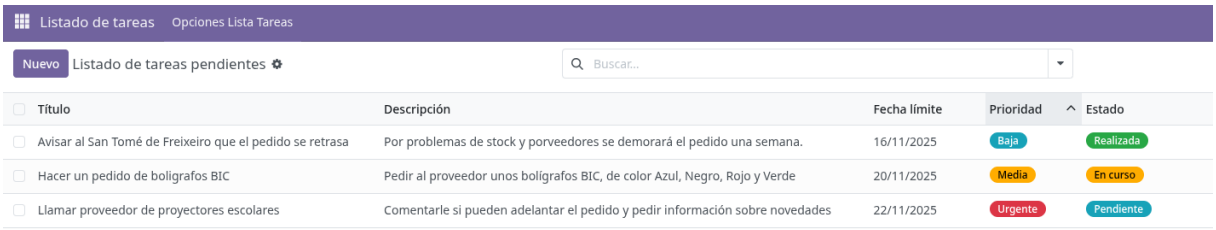

Figura 26: Resultado de nuestras modificaciones de nuestro módulo de Odoo

Así nos quedó nuestro módulo ya modificado, donde podemos ver que nos quedó con estos campos para gestionar nuestras tareas:

- **Tarea** para el nombre de la tarea
- **Descripción** de nuestra tarea
- **Fecha Límite** de la tareas
- La **prioridad** en la que debemos hacer esta tarea (Baja, Media y Urgente)
- Y el **estado** en la que se encuentra la tarea (Pendiente, En curso y Realizada)

6. Conclusión

En esta práctica hemos **aprendido** a cómo **crear y configurar nuestros módulos en Odoo desde cero**, empezando desde un muy simple a otro en el que usamos `__manifest__.py`, `models.py` y `views.xml` para crear un módulo funcional de gestion de tareas.